

# 分块

2019年9月23日 20:50

一. 莫队算法: 适用于离线可加减数据的区间询问,  $O(n \sqrt{n})$

1. 按左端点分 $\sqrt{n}$ 个块, 块内按右端点升序 (或按块号奇偶性分别升序降序)
2. 排序完每个询问后, 根据上次询问的lr差, 增减部分数据, 离线统计进答案
3. 例题: CF86D, 区间内各数据\*数据出现次数的平方, 由平方差公式可知每次只需加减该数据\* $(2*次数+1)$

```
int blk[MN]; //各序号分进的块
struct node{
    int l,r,i;
    bool operator<(const node&t){
        return blk[l]==blk[t.l]&& (blk[l]&1? r<t.r: r>t.r) || blk[l]
        <blk[t.l];
    }
}ask[MN];
int a[MN],cnt[MN<<2];
ll ans,o[MN];

void update(int x,int d=1){    //d=-1时是减少一个x, 否则是增加一个
    if(d==1){
        ans+=(cnt[x]<<1|1)*x;
        ++cnt[x];
    }else{
        --cnt[x];
        ans-=(cnt[x]<<1|1)*x;
    }
}

int n,t; scanf("%d%d",&n,&t);
int sq=sqrt(n);
for_(i,1,n) blk[i]=i/sq;
for_(i,1,n) scanf("%d",&a[i]);    //注意这里空开了下标0
for_(i,0,t) scanf("%d%d",&ask[i].l,&ask[i].r), ask[i].i=i;
sort(ask,ask+t);
int pl=1, pr=0;    //前一个询问的[l,r]
for_(i,0,t){
    const node &nd=ask[i];
    int id=nd.i;
    if(nd.l==nd.r){ o[id]=a[nd.l]; continue; }
    if(pl>nd.l) for_(j,nd.l,pl) update(a[j]);
    else for_(j,pl,nd.l) update(a[j],-1);
    if(nd.r>pr) for_(j,pr+1,nd.r) update(a[j]);
    else for_(j,nd.r+1,pr) update(a[j],-1);
    pl=nd.l, pr=nd.r;
    o[id]=ans;
}
for_(i,0,t) printf("%lld\n",o[i]);
```